Using graph theory to understand interaction patterns in neural data

Danielle S. Bassett[1,2]

[1]*Department of Bioengineering, University of Pennsylvania, Philadelphia, PA, 19104 and*

[2]*Department of Electrical and Systems Engineering,*

University of Pennsylvania, Philadelphia, PA, 19104

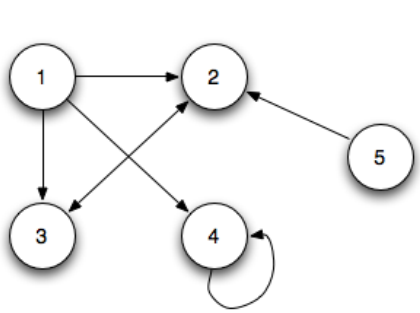
(Dated: October 12, 2016)

Abstract

In this set of three brief sessions, we will address the general question of how to use tools from graph theory and network science to identify, characterize, and interpret interaction patterns in neural data. Interaction patterns are – exactly what they sound like – patterns of relationships between neural elements. These elements may be neurons, other cell types, cortical columns, or large-scale pieces of cortical tissue. The interactions between elements can be defined by synapses, statistical similarities in time series, or shared gene expression profiles, to name a few. The pattern of interactions between elements can be represented mathematically as a graph in which elements are referred to as nodes, and interactions are referred to as edges. With this representation, we can use tools from graph theory to better understand the interaction pattern. After completing these sessions, you will be able to use tools from graph theory to (i) quantitatively characterize local network structure of neural data, (ii) apply graph-based clustering tools to identify modules in neural data, and (iii) quantitatively describe how interaction patterns (networks) change using recent dynamic extensions of graph theory. These tools are generally applicable across many different scales of neural data, from calcium traces or spiking patterns to BOLD or MEG fluctuations, and therefore constitute an important component of the curious neuroscientist's toolkit.

REPRESENTING INTERACTION PATTERNS AS GRAPHS

So much of what we study in neuroscience is inherently *relational data*. While we might not treat it as such, we often have measurements of elements (cells, areas, volumes) that are inherently related to one another based on physical distances, similarities in responses to stimuli, shared developmental factors, or intertwined evolutionary drivers. Yet, despite the pervasive nature of relational data in neural systems, these data have proven difficult to study. Often the challenges principally lie in the fact that relational data are by definition massive multivariate objects. Let's consider n elements that interact with one another, giving $n \times n$ interactions. How do we study this object? How do we characterize it? How do we report its structure? Sure – if there are two elements ($n = 2$), this is tractable. But what happens when $n = 100$, or $n = 1000$, or (even more frighteningly) $n = 100,000$?



	1	2	3	4	5
1	0	1	1	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	0	0	1	0
5	0	1	0	0	0

FIG. 1. **Graphs, adjacency matrices, and relational data.** *(Left)* Relational data can be represented as a graph in which N nodes are connected by E edges. *(Right)* We can represent that graph as an $N \times N$ adjacency matrix $\mathbf{A}$ in which the entry at row i and column j refers to the weight of the edge between node i and node j. From `http://faculty.ycp.edu/~dbabcock/PastCourses/cs360/lectures/lecture15.html`.

Luckily, the field of mathematics offers one natural framework in which to answer these questions, and that is the framework of *graph theory*. A graph $\mathcal{G}$ consists of a set of N nodes $\mathcal{V}$ and a set of $N \times (N-1)$ edges $\mathcal{E}$ that quantify relationships between nodes [1, 2] (see Fig. 1). To tabulate the weight of edges, one can construct an $N \times N$ adjacency matrix $\mathbf{A}$ in which the entry at row i and column j refers to the weight of the edge between node i and node j.

What are some interesting questions that we can ask of such a graph? Well, we could ask about local connectivity rules, coarse-grained connectivity structure, and connectivity dynamics. For each question, let's cover background, mathematics, code, and interpretation.

TASK 1: IDENTIFY AND CHARACTERIZE LOCAL NETWORK STRUCTURE IN NEURAL DATA

Background. We will begin at the smallest scale, and ask by what rules a node might connect to other nodes? While there are many potential ways to ask this question (e.g., see [3] and [4, 5] for two recent and complementary examples), one simple method that comes to mind is to ask whether a node tends to connect to other nodes that connect with each other. In essence, we are asking a question about transitivity. A natural way to answer that question is to count the proportion of triangles (sets of 3 nodes and 3 edges) relative to connected triples (sets of 3 nodes with 2 edges).

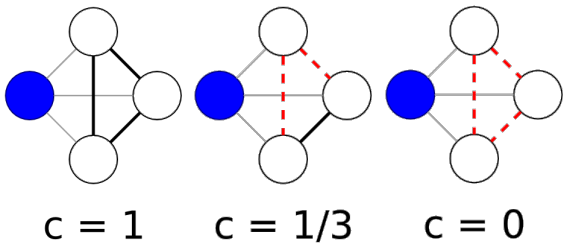


FIG. 2. **Local network structure in neural data.** *(Left)* In a graph where all nodes are connected to one another, the clustering coefficient is equal to unity. *(Middle)* If we delete the edges indicated by the red dashed lines, now only one of the three possible triangles exist, and therefore the clustering coefficient is equal to one third. *(Right)* Now if we delete the red edges, zero of the three possible triangles exist, and therefore the clustering coefficient is equal to zero. From `http://tberg.dk/bl-content/uploads/clusteringcoefficientexample_0_o.png`.

Mathematics. In graph theory, this proportion is called the clustering coefficient [6] (see Fig. 2). Mathematically, we can define it in this way: $C_i = \frac{2|E(N_i)|}{k_i(k_i-1)}$, where k_i is the number of edges emanating from node i, N_i is the set of nodes that node i connects to, and $E(N_i)$ denotes the set of edges with both nodes in N_i.

Let's code it! Now that we understand what a clustering coefficient is, let's calculate it on some real data. First, we'll open up the adjacency matrices that we have been given, which represent functional connectivity (estimated by a wavelet coherence in task-evoked BOLD in the frequency band 0.06–0.12 Hz) between 112 cortical and subcortical brain areas (defined by the Harvard-Oxford atlas) [7].

```
M = load(`matrices.mat');
```

Let's start with the first matrix.

```
matrix1 = squeeze(M.matrices(:,:,1));
```

Calculate the clustering coefficient of the matrix using the Brain Connectivity Toolbox [8], available online at `https://sites.google.com/site/bctnet/` .

```
C = clustering_coef_wu(matrix1);
```

What is the distribution of clustering coefficients over nodes in the network?

```
figure; hist(C);
```

What is the average clustering coefficient for the whole network?

```
C_global = mean(C);
```

Do any brain regions have more or less clustering than expected? To answer this question, we need to define a null hypothesis. Let's decide that our null hypothesis is that all brain regions connect to one another with probability, p, a property that is characteristic of a random (or so-called Erdos-Renyi) graph? To test whether brain graphs show the same properties as graphs under the null hypothesis, we must construct a random matrix, and recalculate the clustering coefficient.

```
uppertriangle = find(triu(matrix1,1)>0);
matrixr = zeros(size(matrix1));
matrixr(uppertriangle) = ...
    matrix1(uppertriangle(randperm(numel(uppertriangle))));
matrixr = matrixr+matrixr';
Cr = clustering_coef_wu(matrixr);
```

Let's look at the distribution of clustering coefficients in the random network overlaid on the distribution from the real network.

```
figure; hist([C'; Cr']');
```

Is this result consistent with our intuition from the matrices themselves?

```
figure; subplot(1,2,1); imagesc(matrix1); subplot(1,2,2); imagesc(matrixr);
```

Interpretation. What does the clustering coefficient mean for brain function? Graphs with high values of clustering coefficients have many strong triangles in the network topology. Intuitively, such a structure is thought to support integrated local processing of information in neural systems [9, 10]. Conversely, graphs with lower values of clustering coefficients have few strong triangles in the network topology, suggesting that they are less well organized to enable local information integration [11]. (Although see [12] and [13] for a broader discussion of interpretational caveats.)

Finally, it is important to note that the mathematical definition of clustering coefficient we discussed earlier is strictly true for binary graphs, where edges are weighted by a value of either zero or one. Yet, the same intuition holds for the calculation of a weighted clustering coefficient [14, 15], which is what we have applied to the weighted matrices in our data [8]. Indeed, in many cases the neural data that we have access to provides information about the strength of a relationship between two entities, not just information about whether a relationship exists or does not exist. When we have this information, it is useful to apply a weighted graph analysis to fully exploit the richness of the empirical data [10].

TASK 2: USING GRAPH-BASED COMMUNITY DETECTION METHODS TO IDENTIFY MODULES IN NEURAL DATA

Background. The clustering coefficient accesses information about the (topologically) local interactions between nodes in a graph. However, one might also be interested in larger scale structure. For example, are there groups of nodes that are densely interconnected with each other, but not necessarily densely interconnected with other groups? This question is naturally answered by *community detection techniques* which search for communities of densely interconnected nodes in graphs [16–18] (see Fig. 3).

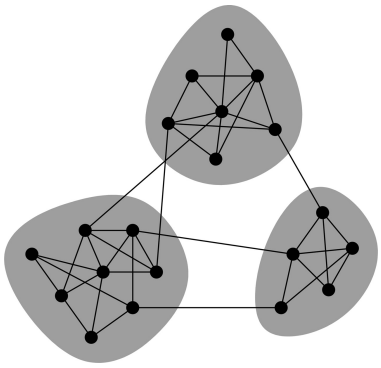


FIG. 3. **Community structure in neural data.** Community detection techniques search for groups of nodes in a network where the nodes in a group are more densely connected to nodes in their own group than to nodes in other groups [17, 18]. From `http://www.pnas.org/content/103/23/8577/F1.large.jpg`.

Mathematics. In graph theory, a popular technique is modularity maximization [19], where a modularity quality function is maximized to identify a partition of nodes into communities. A common modularity quality function is $Q = \sum_{ij}(A_{ij} - \gamma P_{ij})\delta(g_i, g_j)$, where A_{ij} is the ij^{th} element of the adjacency matrix, γ is the resolution parameter, P_{ij} is the ij^{th} element of a null model matrix, Q is the modularity index, $g_i \in [1, \ldots, K]$ indicates to which of K communities node i is assigned, and $\delta(\cdot\cdot)$ is the Kronecker delta function and is equal to 1 if its arguments are the same and 0 otherwise. We can maximize this quality function using a heuristic algorithm [20] to partition nodes into communities (see [21] for best practices).

Let's code it! Now that we understand what a network community is, and what modularity maximization is, let's find some network communities in real data. Again, we will use the Brain Connectivity Toolbox [8], available online at `https://sites.google.com/site/bctnet/`. The inputs to this algorithm are 3-fold: the adjacency matrix $\mathbf{A}$, the resolution parameter γ whose default value is unity, and the initialization community structure (where each node is in its own community).

```
n = numel(matrix1(:,1));
[partition1, Q1] = community_louvain(matrix1,1,randperm(n))
```

Do we get the same answer every time?

```
[partition2, Q2] = community_louvain(matrix1,1,randperm(n))
[partition3, Q3] = community_louvain(matrix1,1,randperm(n))
[partition4, Q4] = community_louvain(matrix1,1,randperm(n))
[partition5, Q5] = community_louvain(matrix1,1,randperm(n))
[partition6, Q6] = community_louvain(matrix1,1,randperm(n))
```

No, and why? Well, modularity maximization is NP-hard, and we only have heuristic algorithms to get us close to the answer (for a thorough description of the near-degeneracy of the modularity landscape, see [22]). Are the values of the modularity index close? and are the partitions similar?

```
Q_values = [Q1 Q2 Q3 Q4 Q5 Q6]
partitions = [partition1 partition2 partition3 partition4 partition5 ...
    partition6];
figure; imagesc(partitions);
```

Yes, the values of Q are similar, and the partitions of brain regions into clusters are similar as well. Are the values of Q or the partitions different from what you would expect in a random matrix? Let's do the same calculations on the variable *matrixr* that we made in **Task 1**.

```
[partitionr1, Qr1] = community_louvain(matrixr,1,randperm(n));
[partitionr2, Qr2] = community_louvain(matrixr,1,randperm(n));
[partitionr3, Qr3] = community_louvain(matrixr,1,randperm(n));
[partitionr4, Qr4] = community_louvain(matrixr,1,randperm(n));
[partitionr5, Qr5] = community_louvain(matrixr,1,randperm(n));
[partitionr6, Qr6] = community_louvain(matrixr,1,randperm(n));
Qr_values = [Qr1 Qr2 Qr3 Qr4 Qr5 Qr6]
partitionsr = [partitionr1 partitionr2 partitionr3 partitionr4 ...
    partitionr5 partitionr6];
```

First notice that the modularity values of the random graph are much lower than that observed in your real data. This indicates that your real data has significant modularity (which can be proven with appropriate statistics). Next, notice that the partitions in your real data are quite different than the partitions in your random data:

```
figure; subplot(1,2,1); imagesc(partitions); subplot(1,2,2); ...
    imagesc(partitionsr);
```

Let's spot check how this maps on to the adjacency matrix representing the relationships between brain regions (thanks to Rick Betzel for the code you are about to use!):

```
[indSort,CiSort] = fcn_order_partition(matrix1,partition1);
[x,y] = fcn_grid_communities(CiSort);
figure; imagesc(matrix1(indSort,indSort)); hold on; plot(x,y,`k');
```

And we can compare to what we would see in the random graph like this:

```
[indSort,CiSort] = fcn_order_partition(matrixr,partitionr);
[x,y] = fcn_grid_communities(CiSort);
figure; imagesc(matrixr(indSort,indSort)); hold on; plot(x,y,`k');
```

For more on "How to", check out [21, 23].

Interpretation. So how do we interpret this analysis? Well, graphs with high values of modularity (Q) have stronger communities than graphs with low values of modularity (assuming the average edge strength and density is the same). Intuitively, such community structure is thought to support a natural balance between information integration within communities, and information segregation between communities [24, 25]. It is also thought to facilitate adaptability over a range of time scales: from short (e.g., learning [26, 27]) to medium (e.g., development [28, 29]) to long (e.g., evolution [30, 31]).

And how do we interpret the partition? Well, nodes that are assigned the same value in the partition vector are assigned to the same community, and therefore are determined to have dense connectivity with one another. We can study all of the nodes that are in community 1 and ask the question of whether they are located close to one another in the brain [32], or whether they span the two hemispheres [33], or whether they code for a similar cognitive function [34]. We can do the same for the nodes that are assigned to community 2; do those nodes code for a different function than the nodes in community 1?

TASK 3: USE DYNAMIC EXTENSIONS OF THESE TOOLS TO DESCRIBE HOW MODULES CHANGE OVER TIME

Background. Understanding how neural system separate interactions into groups that might perform different functions before transmitting that information to another group is important across many scales of neuroscience: from cells to tissue volumes. While the efforts in **Task 2** identified communities in one graph, they do not tell us how communities might change over time – for example, as different types of information are processed, as different stimuli are presented to the animal or human, or as the organism experiences different environments. To address the question of module change, we can turn to multilayer modularity – which is a fairly simple extension of the previous ideas and methods [23].

Mathematics. Mathematically, what we are going to do is string the matrices in a line (for an ordinal system such as occurs when we examine graphs as a function of time), and couple them with identity links weighted by a constant value ω (see Fig. 4). For the requisite equations, and for non-ordinal formulations, check out [36].

Let's code it! Now that we know what multilayer modularity is, let's calculate it on

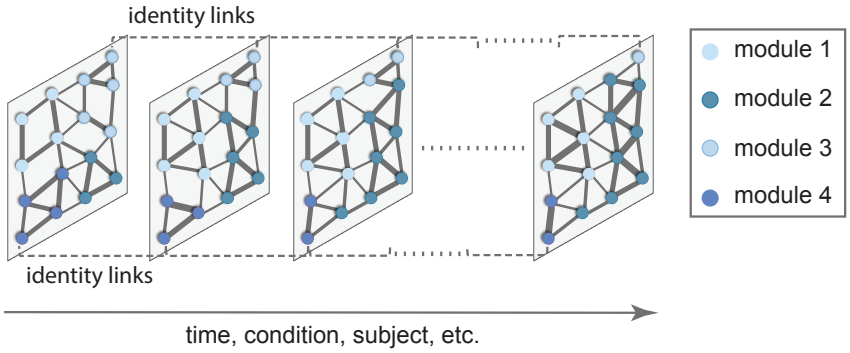


FIG. 4. **Multilayer network representations of neural data.** Patterns of interconnectivity in neural systems can change over time, conditions, subjects, etc. [35]. To address this complexity, multilayer networks can be constructed in which networks in one layer are explicitly linked to networks in other layers by so-called *identity links*. These identity links enable the network to be studied as a single entity, rather than an ensemble, significantly simplifying the mathematics and remaining true to the inherent dependencies between layers.

some real data. In the original data file, we have 9 adjacency matrices. And in our fictitious world, let's assume that those 9 instances are ordered in time. Then, we can identify network communities as a function of time using this multilayer extension of the modularity maximization approach (thanks to Peter Mucha's Netwiki, funded by the NSF and available online here: `http://netwiki.amath.unc.edu/GenLouvain/GenLouvain`):

```
gamma = 1;
omega = 1;
T=length(squeeze(M.matrices(1,1,:)));
B=spalloc(n*T,n*T,n*n*T+2*n*T);
twomu=0;
for s=1:T
    k=sum(squeeze(M.matrices(:,:,s)));
    twom=sum(k);
    twomu=twomu+twom;
```

```
    indx=[1:n]+(s-1)*n;
    B(indx,indx)=squeeze(M.matrices(:,:,s))-gamma*k'*k/twom;
end
twomu=twomu+2*omega*n*(T-1);
B = B + omega*spdiags(ones(n*T,2),[-n,n],n*T,n*T);
[S,Q] = genlouvain(B);
Q = Q/twomu;
partition_multi = reshape(S,n,T);
```

What does the partition of nodes into communities look like?

```
figure; imagesc(partition_multi);
```

Well, it looks like there are about 4 communities, that are expressed very consistently over time. What happens if we tune the value of ω down to 0.2? ((re-run the above with $\omega = 0.2$)). Well, we see that there is more variability in the partitions from time step to time step. This is a key feature of the algorithm: it can be used to probe slow time scale features of your data (using large values of ω) and fast time scale features of your data (using small values of ω). What happens if we tune the value of γ up to 1.2? ((re-run the above with $\gamma = 1.2$)). We see that instead of 5 communities, we now get 28 communities. This is a second key feature of the algorithm: it can be used to probe community structure at small topological scales (using large values of γ that partition the network into many communities) and large topological scales (using small values of γ that partition the network into few communities). For information on data-driven approaches to identifying significant time scales or topological scales in your data, see [21].

Interpretation. What does it mean to have communities change (or remain stable) over time? These (or similar) tools have been used to understand stability in social systems [37], instability in financial markets [38, 39], the history of politics [36], patterns of human behavior [40], and the compression of materials [41]. In the context of neural systems, they offer a principled means of understanding how meso-scale structures in relational data extracted from neural systems may change over time, over animals, over experimental conditions, or over imaging modalities. For two brief reviews, see [23] and [35].

[1] Bollobas, B. *Graph Theory: An Introductory Course* (Springer-Verlag, New York, Heidelberg, Berlin, 1979).

[2] Bollobas, B. *Random Graphs* (Academic Press, 1985).

[3] Klimm, F., Bassett, D. S., Carlson, J. M. & Mucha, P. J. Resolving structural variability in network models and the brain. *PLoS Comput Biol* **10**, e1003491 (2014).

[4] Vertes, P. E. *et al.* Simple models of human brain functional networks. *Proc Natl Acad Sci U S A* **109**, 5868–5873 (2012).

[5] Betzel, R. F. *et al.* Generative models of the human connectome. *Neuroimage* **124**, 1054–1064 (2016).

[6] Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998). 0803.0939v1.

[7] Bassett, D. S., Yang, M., Wymbs, N. F. & Grafton, S. T. Learning-Induced Autonomy of Sensorimotor Systems. *Nature neuroscience* **18**, 744–751 (2015). arXiv:1403.6034v1.

[8] Rubinov, M. & Sporns, O. Weight-conserving characterization of complex functional brain networks. *Neuroimage* **56**, 2068–2079 (2011).

[9] Bassett, D. S. & Bullmore, E. Small-world brain networks. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry* **12**, 512–23 (2006).

[10] Bassett, D. S. & Bullmore, E. T. Small-world brain networks revisited. *Neuroscientist* **Epub ahead of print** (2016).

[11] Achard, S., Salvador, R., Whitcher, B., Suckling, J. & Bullmore, E. T. A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs. *Journal of Neuroscience* **26**, 63–72 (2006).

[12] Rubinov, M. & Bassett, D. S. Emerging evidence of connectomic abnormalities in schizophrenia. *J Neurosci* **31**, 6263–6265 (2011).

[13] Bassett, D. S. & Lynall, M.-E. chap. Network methods to characterize brain structure and function (MIT Press, 2014).

[14] Saramaki, J., Kivela, M., Onnela, J. P., Kaski, K. & Kertesz, J. Generalizations of the clustering coefficient to weighted complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* **75**, 027105 (2007).

[15] Muldoon, S. F., Bridgeford, E. W. & Bassett, D. S. Small-world propensity and weighted brain networks. *Sci Rep* **6**, 22057 (2016).

[16] Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* **69**, 1–15 (2004). 0308217.

[17] Fortunato, S. Community detection in graphs. *Physics Reports* **486**, 75–174 (2010). 0906.0612.

[18] Porter, M. a., Onnela, J.-P. & Mucha, P. J. Communities in Networks. *American Mathematical Society* **56**, 0–26 (2009). 0902.3788.

[19] Newman, M. M. E. Modularity and community structure in networks. *Proceedings of the National Academy of ...* **103**, 8577–82 (2006). 0602124.

[20] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of community hierarchies in large networks. *Networks* **2008**, 1–6 (2008). arXiv:0803.0476v1.

[21] Bassett, D. S. *et al.* Robust detection of dynamic community structure in networks. *Chaos (Woodbury, N.Y.)* **23**, 013142 (2013). 1206.4358.

[22] Good, B. H., de Montjoye, Y. A. & Clauset, A. Performance of modularity maximization in practical contexts. *Phys Rev E Stat Nonlin Soft Matter Phys* **81**, 046106 (2010).

[23] Betzel, R. F. & Bassett, D. S. Multi-scale brain networks. *Neuroimage* **Commissioned** (2016).

[24] Sporns, O. & Betzel, R. F. Modular Brain Networks. *Annual Review of Psychology* **67**, 613–640 (2016). 15334406.

[25] Meunier, D., Achard, S., Morcom, A. & Bullmore, E. Age-related changes in modular organization of human brain functional networks. *Neuroimage* **44**, 715–723 (2009).

[26] Bassett, D. S. *et al.* Dynamic reconfiguration of human brain networks during learning. *Proc Natl Acad Sci U S A* **108**, 7641–7646 (2011).

[27] Bassett, D. S. *et al.* Task-based core-periphery organization of human brain dynamics. *PLoS computational biology* **9**, e1003171 (2013). arXiv:1210.3555v2.

[28] Gu, S. *et al.* Emergence of system roles in normative neurodevelopment. *Proc Natl Acad Sci U S A* **112** (2015).

[29] Fair, D. A. *et al.* Functional brain networks develop from a "local to distributed" organization. *PLoS Comput Biol* **5**, e1000381 (2009).

[30] Kashtan, N. & Alon, U. Spontaneous evolution of modularity and network motifs. *Proc Natl Acad Sci U S A* **102**, 13773–13778 (2005).

[31] Kirschner, M. & Gerhart, J. Evolvability. *Proc Natl Acad Sci U S A* **95**, 8420–8427 (1998).

[32] Lohse, C., Bassett, D. S., Lim, K. O. & Carlson, J. M. Resolving anatomical and functional structure in human brain organization: identifying mesoscale organization in weighted network representations. *PLoS Comput Biol* **10**, e1003712 (2014).

[33] Doron, K. W., Bassett, D. S. & Gazzaniga, M. S. Dynamic network structure of interhemispheric coordination. *Proc Natl Acad Sci U S A* **109**, 18661–18668 (2012).

[34] Mattar, M. G., Cole, M. W., Thompson-Schill, S. L. & Bassett, D. S. A functional cartography of cognitive systems. *PLoS Comput Biol* **11**, e1004533 (2015).

[35] Muldoon, S. F. & Bassett, D. S. Network and multilayer network approaches to understanding human brain dynamics. *Philosophy of Science* **In Press** (2016).

[36] Mucha, P. J., Richardson, T., Macon, K., Porter, M. A. & Onnela, J.-P. J.-P. Community structure in time-dependent, multiscale, and multiplex networks. *Science (New York, N.Y.)* **328**, 876–878 (2010). 0911.1824.

[37] Palla, G., Barabasi, A. L. & Vicsek, T. Quantifying social group evolution. *Nature* **446**, 664–667 (2007).

[38] Fenn, D. J. *et al.* Dynamic communities in multichannel data: an application to the foreign exchange market during the 2007-2008 credit crisis. *Chaos* **19**, 033119 (2009).

[39] Fenn, D. J. *et al.* Temporal evolution of financial-market correlations. *Phys Rev E Stat Nonlin Soft Matter Phys* **84**, 026109 (2011).

[40] Wymbs, N. F., Bassett, D. S., Mucha, P. J., Porter, M. A. & Grafton, S. T. Differential recruitment of the sensorimotor putamen and frontoparietal cortex during motor chunking in humans. *Neuron* **74**, 936–946 (2012).

[41] Papadopoulos, L., Puckett, J. G., Daniels, K. E. & Bassett, D. S. Evolution of network architecture in a granular material under compression. *Phys. Rev. E* **94**, 032908 (2016).